Contents

List of Tables

List of Figures

~

Chapter 1

Introduction

This document reports on the work done under NASA Cooperative Agreement NCC 2-333 during the period August 1990 through March 1991 . The research was carried out by a team of five Ph.D. candidate students from the Stanford University Aerospace Robotics Laboratory under the direction of Professor Robert H. Cannon, Jr. The goal of this research is to develop and test experimentally new control techniques for self-contained, autonomous free-flying space robots. Free-flying space robots are envisioned as a key element of any successful long term presence in space. These robots must be capable of performing the assembly, maintenance, inspection, and repair tasks that currently require astronaut extra-vehicular activity (EVA). Use of robots will provide economic savings as well as improved astronaut safety by reducing and in many cases eliminating the need for human EVA.

The focus of our work is to develop and carry out a set of research projects using laboratory models of satellite robots and a flexible manipulator. The second-generation space-robot models use air-cushion-vehicle (ACV) technology to simulate in two dimensions the drag-free, zero-g conditions of space. Using two large granite surface plates (6' by 12' and 9' by 12') which serve as the platforms for these experiments, we are able to reduce gravity-induced accelerations to under $10^{-5}g$, with a corresponding drag-to-weight ratio of about 10^{-4}—a very good approximation to the actual conditions in space. The flexible manipulator, also using air-cushion technology, is mounted on a third (4' by 8') granite surface plate.

During this period four Ph.D. theses documenting NASA funded research were published. They include Robert Zanutta's thesis on adaptive control of cooperating manipulators, Ross Koningstein's thesis on cooperative arm object manipulation with a two-armed free-flying robot, and Warren Jasper's thesis on thrusterless robot locomotion control for space applications. These projects were funded entirely by NASA. NASA also partially funded the research published in Celia Oakley's thesis on modelling and end-point control of two-link flexible manipulators. Finally, the lab published non-NASA funded research in a thesis by Brain Anderson on end-point position and force control of a minimanipulator on a flexible-drive manipulator. All five of these theses have been inclosed with this progress report.

Our current work is divided into three major research projects: Global Navigation

and Control of a Free-Floating Robot, Multiple-Robot Cooperation, and Dynamic Payload Manipulation. Each of these projects represents an ongoing experimental PhD thesis.

The Global Navigation and Control project demonstrates simultaneous control of the robot manipulators and the robot base position on the free-flying robot model. This will allow manipulation tasks to be accomplished while the robot body is controlled along a trajectory. This project has been completed and is in the documentation phase.

The Multiple-Robot Cooperation project will demonstrate multiple free-floating robots working in teams to carry out tasks too difficult or complex for a single robot to perform. A third space robot model, identical to the robot fabricated for the Thrusterless Locomotion project, recently has become operational– providing the minimal two robots needed for the multiple-robot research.

The Dynamic Payload Manipulation project seeks to demonstrate control of non-rigid payloads and explore the payload's effects on the dynamics of a manipulator system. This research addresses the fundamental issues involved with manipulating space-born objects that possess sloshing fuel tanks or flexible appendages such as solar arrays.

Also, during this period we are launching two new projects. We have begun an investigation of the application of neural networks to space robotics. We have also begun investigating applications of sensor fusion to increase the capabilities of space robots in unstructured environments.

The chapters that follow give detailed progress and status reports on a project-by-project basis.

1.1 Summary of Progress

- Published four Ph.D. theses documenting our NASA-funded research activity over the last five years.
- Implemented a vision-based global positioning system over our large granite surface plate. The system combines the measurements of three real-time cameras.
- Demonstrated a successful rendezvous and capture of a free-flying object with a mobile, two-cooperating arm robot. The system includes a "point and click" graphical user interface.
- Demonstrated initial cooperative manipulation with multiple robots under the management of a coordinating agent. This accomplishment utilized our network-shared-memory multiple robot communication architecture on our real-time system.
- Completed the design and construction of experimental hardware to study the control of a dynamic object. Conducted initial experiments demonstrating that dynamic objects seriously degrade the performance of non-colocated control systems.
- Launched investigations into two major topics: the application of neural networks to space robotics, and the utilization of sensor fusion to increase robustness in unstructured environments.

~

Chapter 2

Autonomous Navigation and Control of Multi-Manipulator, Free-Flying Space Robots

Marc Ullman

2.1 Introduction

This chapter summarizes the progress to date in our research on autonomous navigation and control of multi-manipulator, free-flying space robots. This work represents one of the key elements of our comprehensive effort in developing new technology for space automation. Ultimately, we envision groups of fully-self contained mobile robots making up the core work force in space.

2.1.1 Motivation

Although space presents us with an exciting new frontier for science and manufacturing, it has proven to be a costly and dangerous place for humans. Space is therefore an ideal environment for sophisticated robots capable of performing tasks that currently require the active participation of astronauts.

While earth based robots have not always proved to be cost effective solutions to manufacturing inefficiencies (due to the abundance of cheap labor), the tremendous cost associated with putting humans in space, especially when EVA is required, makes the economics of robots in space particularly attractive.

As our presence in space expands, we will need robots that are capable of handling a variety of tasks including routine inspection and maintenance as well as unforeseen servicing and repair work. These tasks could be carried out by a fleet of free-flying space robots equipped with a set of dextrous manipulators. Such robots must be able to navigate to a job site, rendezvous with the object in need of service, perform the necessary repair

operations, and return to base. Recognizing this need for mobility in space, NASA built the Manned Maneuvering Unit (MMU) to enable astronauts to perform these tasks today.

We are attempting to show, using a laboratory setting, that the underlying technology exists to turn this goal from a vision into reality—that we can assemble a system capable of demonstrating these ideas in a realistic and convincing manner.

2.1.2 Research Goals

The immediate goals of this project are to:

- demonstrate the ability to simultaneously control robot base position and manipulator motions so that a free-flying robot can navigate to a specified location in space while utilizing its arms.
- demonstrate the ability to rendezvous with, capture, and manipulate a free-flying, spinning target.
- provide a high level user interface that enables an operator to control the system by issuing task level commands.
- imbue the system with sufficient intelligence that it can carry out such commands free of any additional operator assistance.
- provide a suitable platform for the eventual addition of A.I. based path planning and obstacle avoidance algorithms which will enhance the robustness of task execution.

2.1.3 Background

Our laboratory work involves the use of a model satellite robot which operates in two-dimensions using air-cushion technology. We have developed a set of satellite robots that, in two dimensions, experience the drag-free and zero-g characteristics of space. These robots are fully self-contained vehicles with on-board gas supplies, propulsion, electrical power, computers, and vision systems. They are also equipped with a pair of two-link manipulators that enable them to capture and manipulate target objects.

Our work emphasizes the modeling of robot dynamics and the development of new control strategies for dealing with problems of:

- a non-inertially fixed base (i.e. free-floating base)
- redundancy with dissimilar actuators
- combined linear and non-linear actuators
- highly non-linear dynamics
- unstructured environments

It also presents a number of challenging system's design problems resulting from the need to carefully integrate many complex subsystems. These include design and construction of the robot itself which incorporates, electrical, gas, sensor, actuator, computer, communication, and vision systems into an autonomous package measuring under 0.5m in diameter by .75m high[1]. Built on top of this hardware platform is a complex computer system architecture consisting of both on-board and off-board processors that communicate via a fiber optic-based Ethernet link. These computers all run a real-time multitasking operating system and perform a variety of sensor and control tasks including real-time vision processing, dynamics computations, closed-loop digital control, as well as high-level strategic control functions including path planning, sensor fusion, and user interface functions.

2.2 Summary of Progress

The following advances have been achieved during the past report period:

- We have demonstrated the successful tracking and capture of a free-flying, spinning object. The object can be initially out of reach of the robot in which case it will first plan and execute an appropriate intercept trajectory for rendezvousing with the object.
- We have added the ability to control the manipulator endpoints (and the object they are grasping) in both global (inertial) and local (base relative) reference frames.
- We have implemented a point and click graphical user interface that enables a remote operator to control the robot by issuing task level commands. This interface allows a complete operation to be specified before it takes place thereby keeping the user "out of the loop."
- We have enhanced our trajectory generation algorithms with such features as calculating the maximum acceleration and making sure that it does not exceed a predetermined limit.
- We have cleaned up and sped up our thruster mapping code.
- We have added some additional consistency checks to our vision system software to help prevent it from mis-identifying target marker patterns.
- We have made minor hardware enhancements that facilitate monitoring of system operation.

2.3 Capturing a Free-Flying Target

We have now successfully demonstrated the tracking and capture of a free-flying, spinning target. The target can be initially out of the robot's reach in which case it will devise an

[1] Not including the camera boom or the manipulators

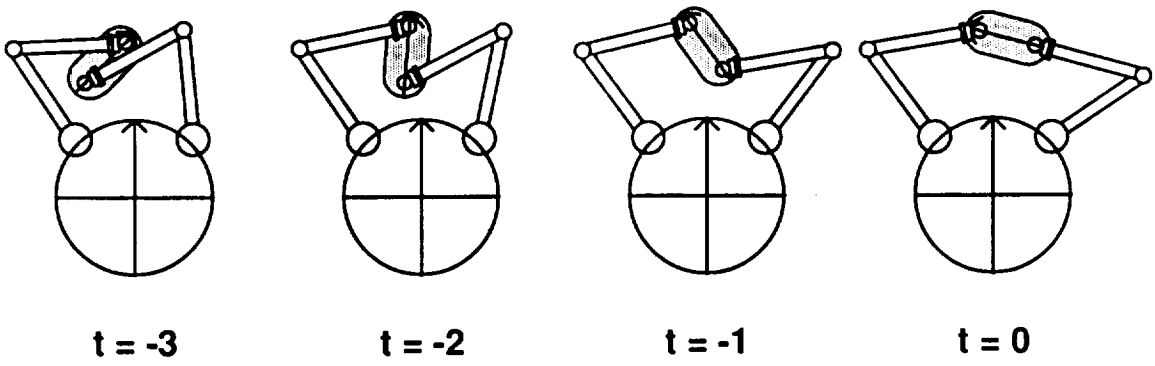

Figure 2.1: Grasp strategy employed when intercepting target employs "look ahead" feature to maximize potential tracking time.

intercept trajectory based on a prediction of where the target will be at some time in the future. The global vision system is used to determine the current position and velocity of the target. Once a feasible intercept trajectory that will take the robot to the object has been computed, the robot begins to execute it, also using the global vision system to monitor its own position. The trajectory is updated once every two seconds to compensate for any unmodeled disturbances that might affect the predicted target location. When the target finally comes into view of the on-board vision system, the strategic control system proceeds to compute closing trajectories that cause the motions of the robot manipulators to correspond with the positions, velocities, and accelerations of the target grip points. These trajectories are designed to intercept the object in such a manner as to allow the maximum time to track and de-spin the object as shown in figure below. Upon completing these intercept trajectories, the system enters a tracking mode utilizing a PID error law to drive the residual grip misalignment error to zero. Once this misalignment error falls below a threshold (currently set at 5mm), the grippers close and grasp the object. A deceleration trajectory is then planned and executed that brings the object to rest in the frame of the robot.

The robot can also "stow" the object or place it in natural carrying position. It can then transport the object to a desired location and place it a specified orientation.

2.4 Control in Multiple Reference Frames

When capturing, manipulating, and transporting an object, it turns out to be very useful to operate—that is, specify desired motions—in both the global (inertial) reference frame as well as the local or robot relative (non-inertial) reference frame. For instance, planning an intercept trajectory is something that can be done very naturally in the inertial reference frame since we have a very good model for the free-flying motion of the object. However, once we have captured the object and wish to transport it, it is much more natural to

specify its position and orientation relative to the robot's current position. To this end, we have transformed our original Cartesian space computed torque formulation that was based on errors in the inertial reference frame to one in which the errors can be expressed in the frame of any rigid body in the system.

2.5 Graphical User Interface

The figure below shows a typical screen from our point and click graphical user interface. This SunView application runs on a Sun workstation and communicates with both vision servers as well as the robot via TCP/IP sockets. The vision systems provide continual position updates of the robot and target positions thereby enabling remote operation since the operator does not need to see the actual robot or target. Clicking on either the robot or the target selects that object and makes it active as is indicated by the bold outline. Clicking and dragging an object not only selects it but also produces a ghost image that can be repositioned to a desired location and orientation. For instance the user can click on the object, reposition its ghost image and then click on the "MOVE" button. This sends a message to the robot telling it where to place the object. If it does not currently have possession of the object, the robot will first rendezvous with it and capture it. It will then stow it and transport it to the desired location, and finally rotate it into the requested orientation.

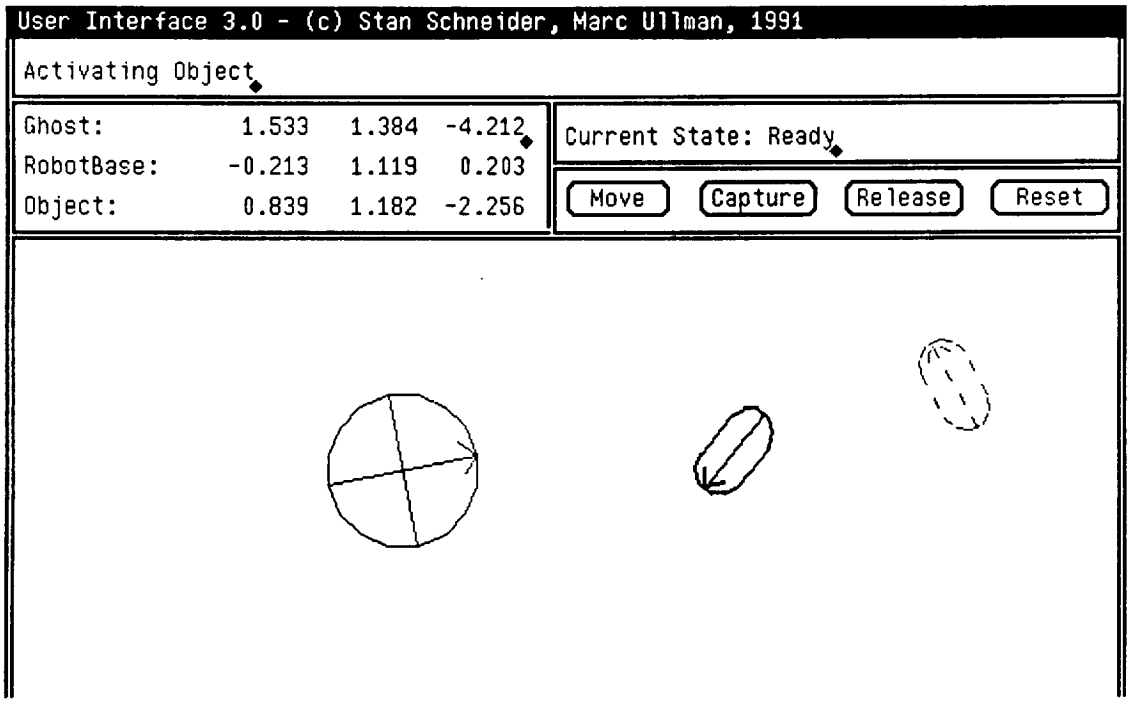

Figure 2.2: A typical view of the graphical user interface.

2.6 Trajectory Algorithm Enhancements

In completing the rendezvous and capture described above, we found it necessary to add a number of new features to our trajectory generation algorithms. These features include:

- The ability to determine the maximum acceleration to be encounted while executing the trajectory.
- The ability to asynchronously query the trajectory for its expected position and velocity at any arbitrary time in the future.
- The ability to compute the minimum time trajectory that will not exceed a specified maximum acceleration.

2.7 Improved Thruster Mapping Code

Our robot is equipped with eight bang-bang gas jet thrusters mounted as four 90 deg pairs at the corners of the robot base. Each axially paired set of thrusters can be thought of as one bi-directional thruster having three possible operating states: forward, off, and reverse. Therefore there are 3^4 or 81 possible thruster configurations. Of these 81 possible combinations, 65 of them result in unique sets of forces and torques–the remaining 16 resulting in duplicates of these. Thus the thruster mapping problem is one of finding the "best" match between the desired set of forces and torques [F_x F_y T_θ] and the 65 possible thruster configurations. To make this comparison consistent along all three axes, we convert the desired forces and torques into their thruster equivalents via the following scaling:

$$\begin{bmatrix} F_x \\ F_y \\ T_\theta \end{bmatrix} \begin{bmatrix} 1/(\text{Force per Thruster}) \\ 1/(\text{Force per Thruster}) \\ 1/(\text{Torque per Thruster}) \end{bmatrix} = \begin{bmatrix} N_x \\ N_y \\ N_\theta \end{bmatrix}$$

Then we search for the minimum norm error between [N_x N_y N_θ] and the list of possible thruster combinations. Since this search is occuring once per sample period, it is imperative that it be fast. We can take advantage of the symmetry properties of the possible thruster configurations. First we recognize that [N_x N_y N_θ] space can be broken in to eight symmetrical octets, differing only in the signs on each term. This reduces our search space from 65 entries down to 16. We can further reduce this by taking advantage of the fact that each octet is symmetrical across the line $N_x = N_y$ which further reduces out list down to 11 entries. Once we have found the best match among these eleven entries by taking the absolute values of [N_x N_y N_θ] we determine the final pattern by looking at the sign of each element and applying the necessary correction. This step is done by using the match number as an index into table which is selected based on the signs of the original forces and torques. Once the final pattern is determined, the corresponding thrusters can be activated.

2.8 Vision System Consistency Checks

We have added several new consistency checks to our vision tracking algorithms in an attempt to further enhance their robustness. These new checks include:

- Continual inter-marker spacing checks. This feature assures us that if an object is incorrectly identified because some extraneous point (e.g. a manipulator endpoint) happens to make the apparent geometry consistent with a desired pattern, it will be rejected as soon as that geometry changes and is no longer consistent with the desired pattern. In the case where we have three points, the algorithm waits until two distances are out of spec so that it knows which point to reject. In the case where only two points are being tracked, the object is considered lost when the inter-marker distance is violated.

- The ability to reject mirrored patterns. In the original version of the vision software, it was possible to "re-find" the third point in such a way that the object's orientation would instantaneously flip by 180 deg. Since this is certainly unrealistic, we now reject a matching third point if it would cause the object's orientation to change by more that 45 deg.

- We now support programmable parameters for both the tracking tolerance—how far a point can move between successive frames—and object identification tolerance—how closely the inter-marker spacing dimensions must be matched.

2.9 Experimental Hardware

This section reviews the latest refinements we have made to our experimental hardware setup. In as much as the hardware is now fully complete, these represent fairly minor improvements.

2.9.1 Analog Multiplexer

We have finally designed and implemented an analog multiplexer that enables us to monitor a number of slowly changing signals with the one remaining channel on our A/D converter. These signals include the positive and negative power bus voltages and the high and low pressure sensors. The channel to be read is selected by writing a channel number out via our digital I/O board.

2.9.2 Battery Sensing/Actuation

We have also finally wired up the capability to switch the two battery packs on and off under computer control. This facility along with the previously mentioned bus voltage sensing allows us to check the battery voltages under load conditions.

2.9.3 Safety Sensing

We have also added the ability to detect whether or not the manual safety override is engaged so that the computer can wait for this signal before proceeding. (The manual safety override disables the manipulator motors and the gas jet thrusters.)

2.10 Future Work

This project is essentially complete now that we have demonstrated the ability to rendezvous with, capture, and manipulate a free-flying, spinning object. At this point the author is working on writing up the results. There are, however, a number of additional steps that could be taken to further this research. These are divided into three categories and summarized below.

2.10.1 Hardware Improvements

- Add force sensors to the grippers. The grippers are designed to accommodate semiconductor strain gages for sensing forces at the tips; however the gages have never been mounted. The requisite electronics and cabling is all in place from the original grippers that were equipped with strain gages.

- Replace the fiber optic communication link with a wireless equivalent. Motorola has recently introduced their Altair wireless Ethernet link that uses microware technology to achieve the full 10 Mbps bandwidth. This product could be readily adapted to our needs—the main drawback being its price of over $7000.

- Compensate for manipulator motor torque non-linearities. The limited angle DC torque motors that we are using to drive the manipulator joints suffer from a torque roll off at large angles (falling to zero at ± 90 deg). This roll off could be compensated for by applying a correction to the requested torque. A fourth-order polynomial would probably do fine.

2.10.2 Control System Issues

- Drop negligible terms from dynamic compensation. Currently, we are using the complete equations of motion in our computed torque or inverse dynamics controller. Because these equations are generated automatically from a system description, the only drawback to using them is the computational power required to evaluate them in real-time. It has been show [7] that a number of these terms are negligible and hence could be eliminated in an effort to speed up the computations.

- Use estimates of actual thruster forces in computed torque controller. Although our thrusters are of the on-off or bang-bang type, our controller assumes that they are proportional devices and requests arbitrary force/torque levels from them. These force/torque demands are met through time averaging but not instantaneously. The

controller could be made more complicated (and less modular) by taking into consideration the discrete force/torque levels that the thrusters are actually able to deliver.

- Examine stability and robustness issues. As with all control systems, the question of stability arises. Little work has been done on trying to prove the stability limits of the current control architecture.

- Use adaptive control to eliminate need for knowing mass properties of object being captured and manipulated. Currently we make use of a priori information about the mass properties of the object we are attempting to capture and manipulate. However, recent advances in adaptive control should allow new controllers to be developed that will operate safely without this advance knowledge. Rather, the controller will determine this information in real-time as it interacts with the target object.

2.10.3 High-Level (AI) Task Planning Issues

- Perform more difficult docking and insertion tasks. This test facility could be used to demonstrate docking and insertion tasks with additional objects that could be tracked and monitored by both the global and local vision systems.

- Use more sophisticated path planning algorithms for rendezvousing with moving objects. We are currently using a rather simple algorithm for intercepting the target if it is initially out of reach. A number of more sophisticated algorithms have been suggested and these could be tested experimentally.

- Introduce obstacles to complicate the path planning problem. In a real world scenario, a robot would likely have to contend with other objects in its workspace. We could explore the added complexities these obstacles would impose by trying out various path planning algorithms that handle both dynamic constraints as well as stationary and moving obstacles.

- Use multiple robots to carry out tasks beyond the capability of a single robot. Manipulation and assembly of large objects requires teams of cooperating robots. The added levels of complexity introduced by coordinating and controlling multiple robots provide a number of new issues that must be handled in order to produce a successful system.

As is shown by the presence of some of the other sections of this report, investigations into a number of these issues are already underway.

Chapter 3

Multiple Robot Cooperation

William C. Dickson

3.1 Introduction

This chapter summarizes our progress to date in the area of multiple robot cooperation. This work will eventually unite the various lines of research presently being conducted in fixed- and floating-base cooperative manipulation, and in global navigation and control of space robots. Our goal is to demonstrate multiple free-floating robots working in teams to carry out tasks too difficult or complex for a single robot to perform. Achieving this cooperative ability will involve solving specialized problems in dynamics and control, high-level path planning, and communication.

3.1.1 Research Goals

Some of the goals of this project are:

- Cooperative manipulation and assembly by multiple robots.
- Fine cooperative manipulation in presence of on-off control.
- Development of control strategies for path following.
- Path generation considering dynamic constraints and obstacle avoidance.

3.2 Progress Summary

Activities completed from September 1990 to February 1991 were:

- Improved momentum wheels operational on robots.
- Improved grippers operational on robots.
- On-board power now available on third second-generation mobile robot.

- Improved methodology for multiple-robot control of manipulation object.
- Coordinator module extended to allow use of user interface.

3.3 Experimental Hardware

3.3.1 Overview

The experimental hardware associated with this research currently consists of an off-board vision system, two mobile robots, an off-board coordinator processor, and a manipulation object. The robots and object use self-powered air bearings for flotation on a 6' by 12' granite table.

3.3.2 Vision System

The vision system consists of a camera mounted above the granite table, an ARL-developed Point Grabber Vision board [2], and a commercial 68030-based computer for vision processing. The vision board converts camera bright spots into a list of pixel coordinates and intensity values. The bright spots are produced by infrared (IR) light emitting diodes (LEDs) located on the robots and object. The vision computer uses the information generated by the point-grabber board to determine the positions of the robots, the robot manipulator endpoints, and the object(s) [3].

3.3.3 Robots

The robots used in this research are nearly identical to the original second-generation robot currently used in the Navigation and Control research. One major differences is that these robots utilize a momentum wheel — allowing the robots to control their orientation without the use of thrusters. Second, unlike the original second-generation robot, these robots currently have no on-board vision system for improved workspace sensing.

Newly designed momentum wheels were recently installed on the robots. The new wheel/motor configuration quadrupled the torque-per-current and maximum torque of the actuator.

The grippers are pneumatically driven plungers used by the robots to manipulate floating objects. The latest model of the grippers, featuring commercial linear bearings, was recently added to the robots.

3.3.4 Coordinator

The coordinator's role is to orchestrate the activities of the robots in response to the inputs of the user. Acting as a protective buffer, the coordinator informs the user if a task cannot be completed as inputted. Also, the coordinator decides which robot or set of robots should be assigned to a particular task, may make choices for the robots where appropriate, and may override ongoing activities of the robots.

3.3.5 Manipulation Object

The manipulation object is constructed from two half-square-feet floating pads connected by a three-feet-long metal bar. Four cylindrical grip ports on the object facilitate grasping by the robots. Battery powered IR LEDs allow the object to be tracked by the vision system.

3.4 Control

The basis of a control scheme that facilitates cooperative manipulation by multiple robots should center on the desired motion of the manipulated object. The operator's concern is the proper positioning of the object— not the control torques and forces on the robots. Following this philosophy, the user should be able to input to the robot system desired object motions at a high level through a user interface. Once the robot system knows the desired motion of the object, joint-robot-level, object-level, and robot-level controllers determine the necessary control torques and forces.

This section discusses the structure of this four-level control hierarchy consisting of the user interface, joint-robot-level control, object-level control, and robot-level control.

3.4.1 User Interface

The user interface is presently the Graphical User Interface developed by ARL, described in the Seventh semi-annual report [3]. This interface allows the user to input high-level commands such as "catch","move", or "insert". The interface, running on a Sun workstation, informs the coordinator processor of the user's commands. Depending on the task, the robots poll the coordinator for new information, or the coordinator sends new data when appropriate.

3.4.2 Joint-Robot-level Control

Robots cooperatively manipulating an object must attain two goals for success: 1) their workspaces must be maintained at relative positions determined by the geometry of the object that they grasp, and 2) they must move such that the object they grasp can be brought to its final desired state. A few definitions of terms will aid in the discussion of joint-robot-level control:

Destination State: Final desired state of manipulated object.

Manipulation Line: Line segment between two robot workspace centers.

Grip Center: Position of point on object midway between two ports gripped by one robot, or position of the port if the robot is gripping one port.

Transition State: State of object when its Grip Centers are on the Manipulation Line and are equidistant from the Manipulation Line midpoint.

The Grip Centers of the object at the Destination State define the final desired positions of the robot workspaces. Until the robot workspaces are within a certain distance of these positions, the object is regulated to the Transition State. During this transition phase, the robots attempt to meet the two object manipulation goals in the following way. The desired workspace position is defined as the Grip Center of the object at the Transition State—ensuring that the robots' workspace separation matches the geometry of the object. The desired workspace velocity is determined by a control law that attempts to drive to zero the error between the Transition State and the Destination State. Presently, this control law is an algorithm that combines saturating values of weighted errors in translation and orientation into the resulting desired workspace velocity.

Once the robot workspaces are within close range of their final positions (as determined by the Destination State), the desired object state is the Destination State. Also, the robots attempt to regulate their workspaces to the Grip Centers at the Destination State rather than the Transition State.

3.4.3 Object-level Control

Each robot determines, using the same algorithm, the desired acceleration of the object that will guide the object to the desired state determined by either the Transition State or the object's Destination State. This algorithm is presently a simple proportional-derivative (PD) controller on the state error. These desired object accelerations are then used to determine the accelerations and forces at the grasp points on the object (the present object has four useable manipulation ports).

3.4.4 Robot-level Control

Each robot has knowledge of how the team of robots is currently grasping the object (the grasp configuration), allowing each robot to determine which sets of grasp port accelerations and forces should be associated with its own manipulator endpoints. The desired workspace position and velocity determined by the joint-robot-level controller are now used to determine controls for the base. Each robot uses the desired accelerations and forces at its endpoints together with the base controls to determine the necessary joint torques, as described in the Seventh semi-annual report [3].

3.5 Experimental Results

Experiments have successfully demonstrated multiple-robot manipulation of a floating object being controlled as described above. In these experiments, an off-board vision system tracks the positions and velocities of the object and two robots (as well as the robots' manipulator endpoints), and sends this information via the network to each of the robots.

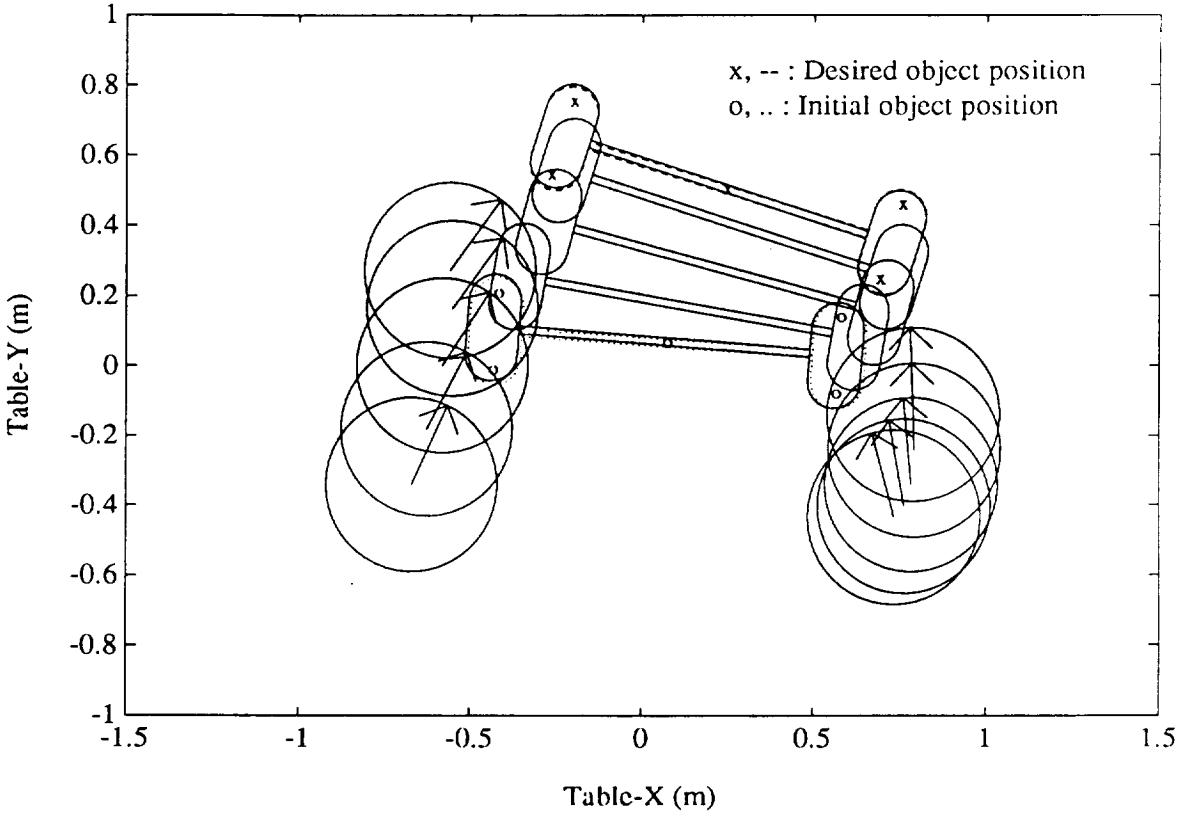

Figure 3.1: Two-Robot Object Manipulation

The robots move the object to a location specified by the operator via a user interface. The on-board air thrusters and momentum wheels control each robot's position and orientation as described. Figures 3.1 and 3.2 show the results of an example object slew. The translational errors were regulated to sub-millimeter levels, while the rotational error was less than one degree. Note that the step changes in the "desired" positions indicate the new Destination State of the object. As previously discussed, the object is regulated to a position near the robots when the robots are out of range of the Destination State. Once in range, the robots regulate themselves and the object to the Destination State. In the example slew, the robots came into range at the 18 second point of the run, or about 14 seconds after the new Destination State was commanded.

3.6 Future Work

Extension to the multiple-robot research include capturing and docking the manipulated object. Currently, the object is always in the grasp of the robots. Also, the robots need to have knowledge of the environment to allow for path planning and obstacle avoidance.

The following hardware issues remain:

- Fit new grippers with force-sensing strain gages for improved control.

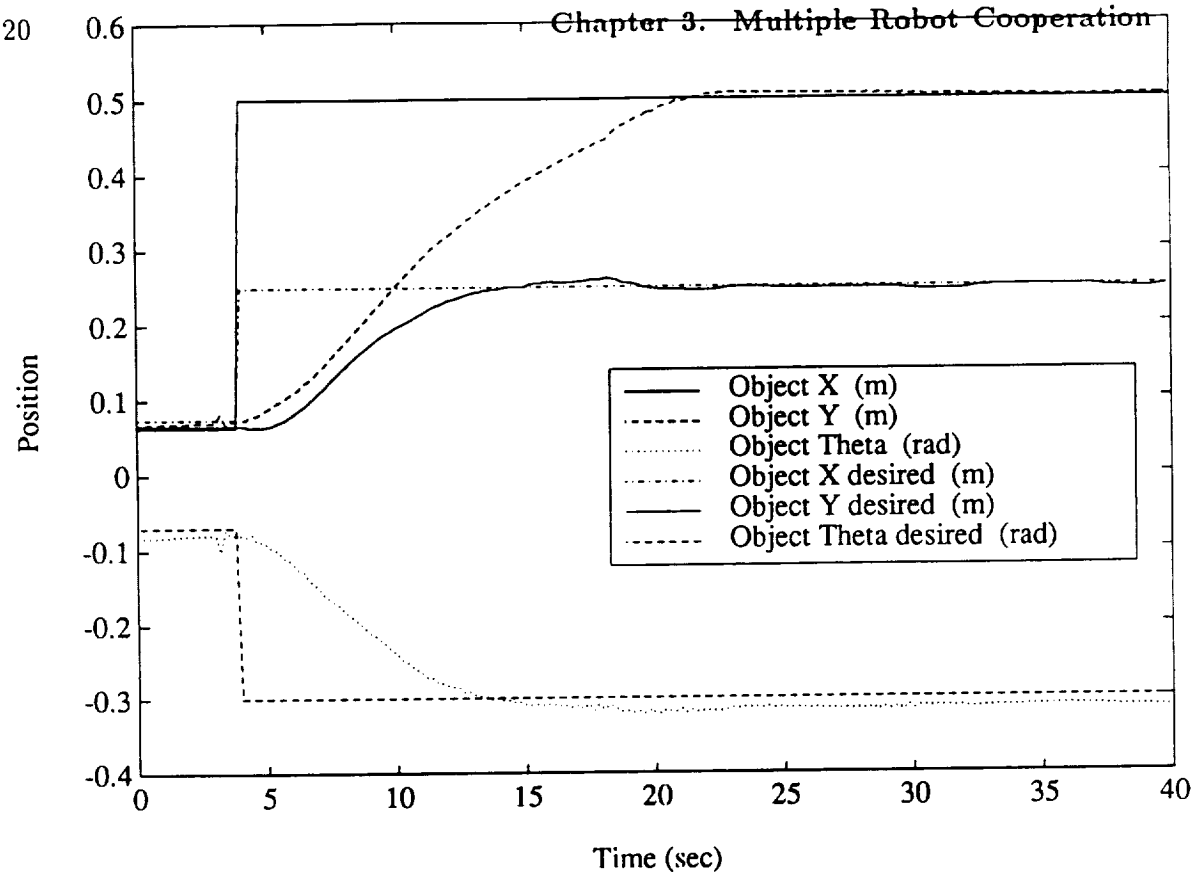

Figure 3.2: Object Slew Performance

- Attain wireless Ethernet communication to replace fiber-optic and coaxial cables.

Chapter 4

Experiments in Control of a Flexible Arm Manipulating a Dynamic Payload

Lawrence J. Alder

Submitted For Presentation ASME 1991 Winter Annual Meeting

4.0.1 Abstract

In many applications flexible robot arms may be handling payloads that cannot be modelled as rigid bodies. In space applications, the RMS (remote manipulator system) will be manipulating satellites that may contain fuel or have flexible appendages [4]. Most high performance control schemes for these flexible manipulators require some form of end point feedback. Such control systems have been shown to be sensitive to unmodelled dynamics in the payload. If the dynamics are not accounted for in the control design, degraded performance and instability are possible.

An experimental apparatus is described that has been constructed for the purpose of investigating the effects of payload dynamics on the control of a flexible robot arm. First, some challenging design goals for the hardware are described. Then, a finite element model of the proposed hardware is developed to aid in the design process. Using the finite element model as a design tool, the actual experimental hardware has been designed and built. The properties of the actual hardware are presented and shown to agree with the predictions.

Finally, a preliminary non-collocated controller which ignores the payload dynamics is presented. The performance of this controller with rigid and dynamic payloads are evaluated. The performance of the controller with the dynamic payload is shown to be inadequate for practical applications, and provides motivation for future study of the problem.

4.0.2 Introduction

Almost all existing control systems for flexible robotics are designed for a payload that can be modelled as a rigid body. It has been shown that high performance non-collocated controllers are sensitive to the mass and inertia of the tip [9]. It has also been shown that a controller that is tunned for a particular tip mass may in fact be unstable for a different tip mass. There have been successful demonstrations of adaptive endpoint control with unknown tip mass. However, this problem becomes even more difficult when the payload is large and must be modelled with a complete inertia matrix. There have been some studies of how a controller might adapt to different payload inertias, but this is still a difficult problem to handle.

In realistic applications, large flexible robots will be required to handle payloads that cannot be modelled with just an inertia matrix. The payloads may have internal dynamics such vibration of solar panels on a small satellite being deployed. Some payloads such as fuel tanks will have internal dynamics that besides being nonlinear are not even modelable as ordinary differential equations. Since it has already been shown that non-collocated control systems are sensitive to the payload inertia properties, it is now important to ask how sensitive are the controllers to the dynamic properties of payloads.

In this paper, it is shown that non-collocated flexible robotic systems are sensitive to the payload dynamics. The design of controllers for this class of payloads is difficult. Most of the internal vibrations or oscillations of the payload cannot be directly sensed or even modelled. An experimental apparatus is required that can be used to evaluate emerging control strategies. A candidate apparatus that has been designed and constructed is described below. Experiments with this apparatus demonstrate that it exhibits the sensitivities that characterize the fundamental problems associated with precision tip position control in the presence of unknown payload dynamics.

4.0.3 Design Objectives-Apparatus Description

This section describes an experiment that highlights the effect of payload dynamics on a flexible manipulator. To make such an experiment interesting, it is desired to have a large amount of coupling between the dynamics of the payload and the manipulator. The experiment should also be as simple as possible while still exhibiting the fundamental issues of a flexible robot arm.

Keeping these goals in mind, a flexible one link planar manipulator has been chosen for the experiment. This allows a comprehensive study of manipulator vibrations without unnecessary complications of out of plain vibrations or multiple link non-linearities. The flex arm has been designed to have three vibration modes below ten hertz when grasping payloads.

In order for the coupling between payload and arm to be significant, the mass of the payload must be large relative to the mass of the beam. This is also a realistic scenario for a shuttle arm grasping a fuel tank. In order to avoid out of plane vibrations while carrying large payloads, the payloads float on an air bearing.

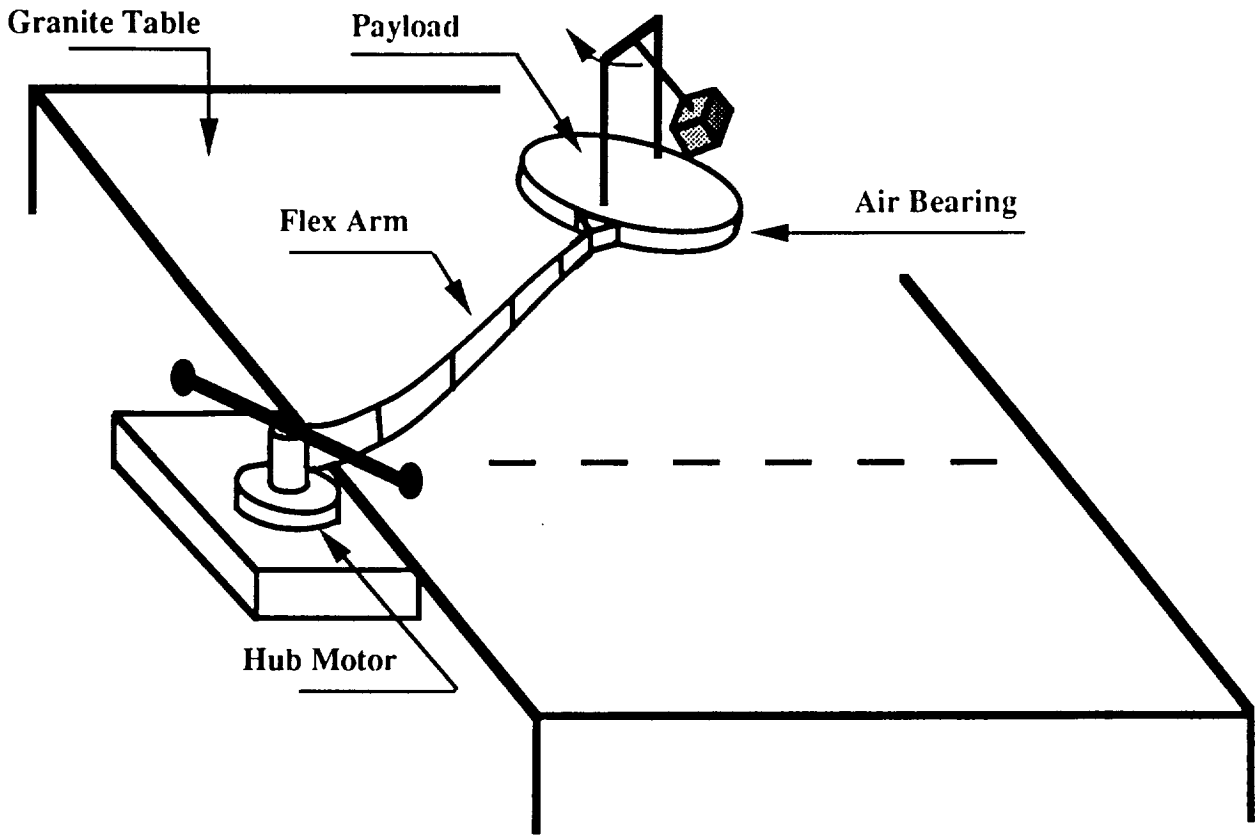

Figure 4.1: **Experimental Apparatus Schematic**

To provide a first order approximation of a sloshing fuel, a pendulum has been chosen for the payload [1]. This allows the study of the effects of payload dynamics on the system without the complexities of modelling a sloshing fluid which could obscure or make difficult the study of some fundamental issues of vibration coupling. In order to provide large dynamic coupling, the payload has a large fraction of its mass belonging to the pendulum. The pundulum is oriented such that it oscillates perpendicular to the axis of the arm. The pendulum frequency can also be varied over a wide range of interest.

In summary, a schematic of the apparatus is shown in Figure 4.1. The figure shows a one link flexible manipulator grasping a payload that floats on an air bearing. The payload has internal dynamics in the form of a pendulum that can oscillate in one degree of freedom (DOF) only.

4.0.4 Parameter Selection

Given the form of the experimental setup as described in the above section, beam parameters such as length, mass, and stiffness and payload parameters such as mass, and inertia must be selected. A complete list of the parameters available as design variables for the experiment is shown in Table 4.1. The parameters are to be selected based on the following

criterion. First, there should be three vibration modes below ten hertz. Second, the payload frequency can be varied from below first mode of arm to above second mode. Finally, the effect of the payload dynamics on the transfer function from hub torque to tip position should be large. This last requirement is there to make endpoint control in the presence of the payload dynamics difficult.

In order to select parameters that meet the above criterion, a mathematical model of the system is necessary. For parameter selection the finite element model has been used. The central limitation (advantage) of the finite element model is that it assumes a completely linear system. The model is formed using the Consistent Mass approach as described in [8]. Linear interpolation functions are used.

Using the finite element model, frequency response plots are generated for different values of the design variables until the design criterion have been met. Table 4.1 shows values of the design variables after several iterations.

Figure 4.2 shows theoretical frequency response plots generated from the finite element model with the parameters shown in Table 4.1. Figure 4.2 shows the static payload case (no internal DOF -pendulum locked) and dynamic payload case (pendulum free to oscillate). One can see that three flexible modes of the beam are below ten hertz. The dynamic payload oscillates at about two hertz. However, by varying the length of the pendulum, the frequency can be varied from nearly one (below first beam mode) to five hertz (above second beam mode). The oscillating payload introduces an extra pole-zero pair in the transfer function from torque to tangential displacement of the tip. Based on the above, the parameters in Table 4.1 meet the design criterion.

Parameter	Value
Length	0.75 m
Beam mass	0.36 kg
Beam stiffness (EI)	0.77 Nm^2
Hub inertia	$1.4x10^{-2}$ kg/m^2
Modal damping	3%
Fixed tip mass	1.2 kg
Fixed tip inertia at cm	$1.0x10^{-2}$ kg/m^2
Oscillating mass	0.5 kg
Oscillating inertia about own cm	$1.7x10^{-4}$ kg/m^2
Oscillating mass damping ratio	.1%
Freq of payload	1-5 Hz

Table 4.1: Beam and Payload Parameters

However, other transfer functions besides those of Figure 4.2 are important. Since the payload has a significant inertia, one must control both the position of the center of mass of the payload, and the orientation of the payload. For payload with inertia, the existence of a non-collocated zero implies that it is possible for the control system to regulate endpoint position while the orientation is oscillating at the frequency of the non-collocated zero.

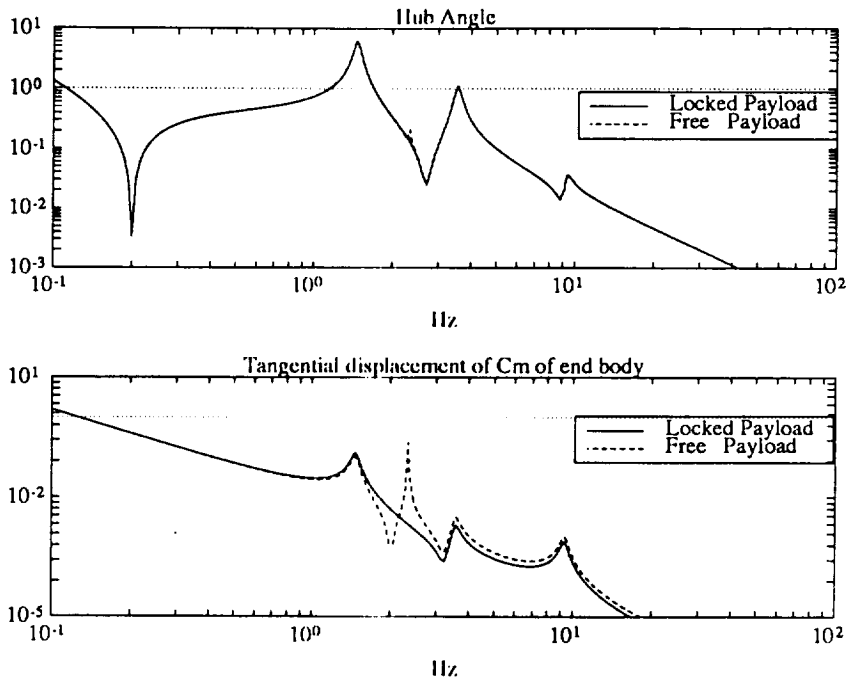

Figure 4.2: **Theoretical Frequency Response**
Generated From Finite Element Model

To prevent this from happening, one must control orientation as well as endpoint position. Figure 4.3 shows the transfer function from hub torque to inertial orientation of the endbody (endbody is any rigid body rigidly attached to the endpoint of the arm). This plot also shows the effect of the pendulum on the transfer function is small.

4.0.5 Nonparametric Plant Identification

This section describes a non-parametric identification of the experimental apparatus. The parameters shown in Table 4.1 have been used to construct the experimental hardware schematically shown in Figure 4.1. Figure 4.4 shows the sensors, actuators, and computers that are available to identify and control the system. To summarize, there is an angle and rate sensor along with a limited angle torquer at the hub. There is a camera that can sense position and orientation of the endbody. Finally, there is a measurement of the position and rate of the oscillating part of the payload. The measurement of the pendulum position and rate is provided only for identification purposes.

Figure 4.5 shows experimental frequency response data. The plots compare well with the finite element predictions in Figures 4.2,4.3. The experimental frequency responses are for the static payload only.

Figure 4.5 also shows a best fit to the frequency response. The best fit is generated

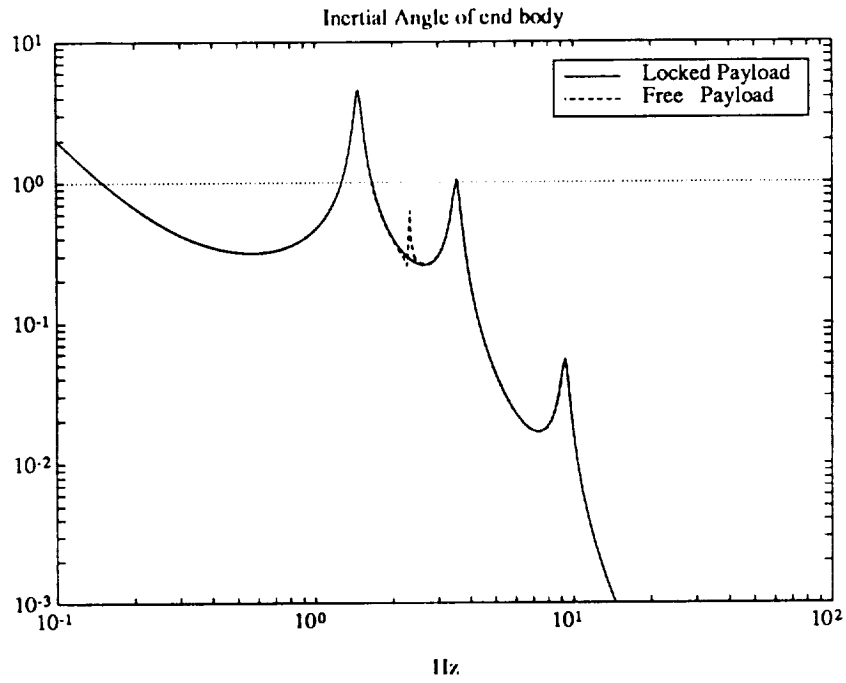

Figure 4.3: **Theoretical Frequency Response**
Generated From Finite Element Model

by assuming a three pole (plus rigid body pole) linear model for the beam, and then by searching in pole zero space to minimize the two norm of the log of the difference between actual data and the model. The best fits represent a linear model that models the actual plant frequency response. This linear model is later used in the design of a controller.

The transfer function from hub torque to tip position is not shown since the gain of that transfer function is very low (see Figure 4.2) and hence the frequency response is hard to obtain experimentally. This implies a controller that relies on this signal to control vibrations may be difficult to implement as will be discussed further in the next section.

4.0.6 Non-collocated Control Without Accounting for a Dynamic Payload

This section presents a preliminary non-collocated controller which ignores the payload dynamics. The controller is experimentally implemented, and is then used to manipulate both static and dynamic payloads. The purpose of this is to show experimentally that the system is sensitive to payload dynamics, and that more sophisticated control strategies will be needed to solve the dynamic payload problem.

Since the goal of the robot is to position the endbody, a controller that feeds back endpoint position should be implemented. However, relying on endpoint position as the only

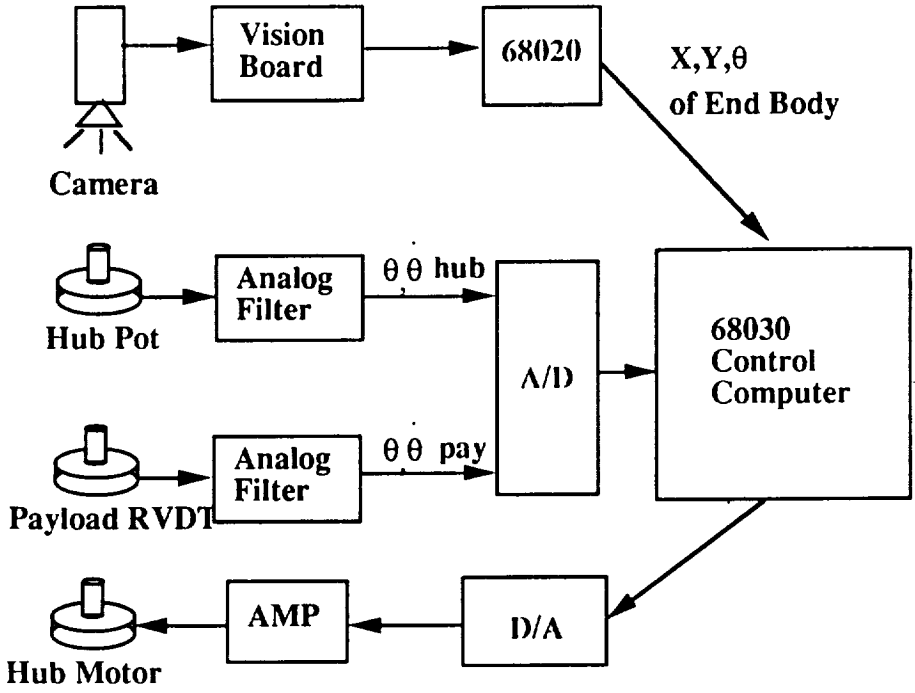

Figure 4.4: I/O Schematic

non-collocated measurement is inadaquate when the payload is massive and has significant inertia. Again referring to Figure 4.2, one notices that not only is the gain on the tip transfer function small, but that there is a non-collocated zero in the tip transfer function at a frequency near a collocated zero at 3Hz. This will make it hard for a controller to identify and reject frequencies near 3 Hz. However, Figures 4.3 and 4.5 show that the transfer function from hub torque to inertial endbody angle does not exhibit either of the above problems.

A non-collocated controller has been designed following the procedures of Schmitz [10]. However, instead of feeding back endpoint position. The endbody angle has been used. This controller is less sensitive to dynamics of the payload than one that feeds back tangential endpoint position, but it does provide a first look at a non-collocated controller trying to control a dynamic payload.

Figure 4.6 shows a block diagram of the control system. The estimator has been designed with the best fit linear models from Figure 4.5 as described in the section on plant identification. The control objective for the LQR is shown in Equations 4.1.

$$J_{lqr} = \sqrt{\theta^2_{endbody} + \frac{1}{2}\dot{\theta}^2_{endbody} + \frac{1}{500}T^2} \tag{4.1}$$

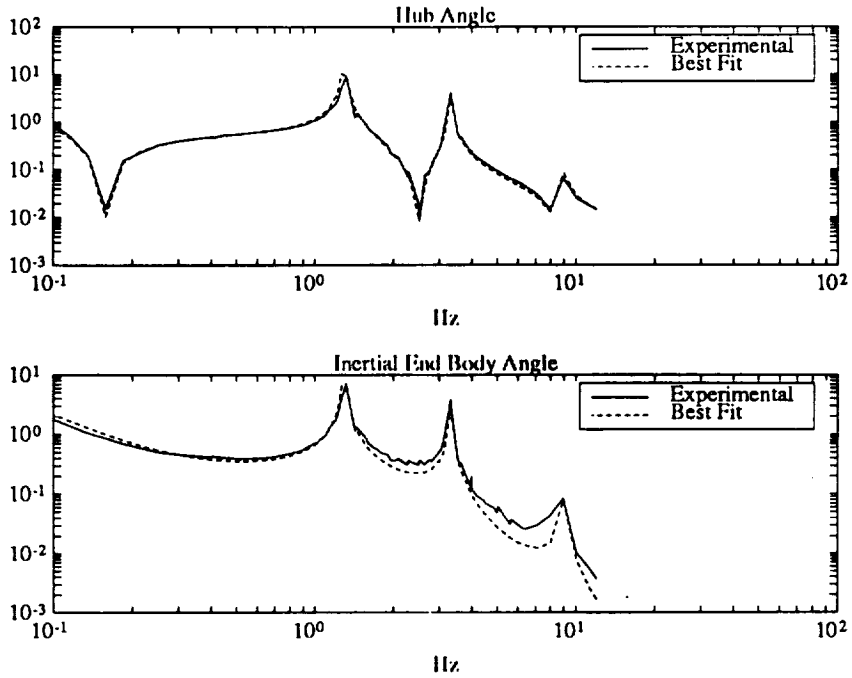

Figure 4.5: **Experimental Frequency Response**

where T = actuator effort

The closed loop performance is first examined with a static payload (pendulum locked). Figure 4.7 shows a step response of the closed loop system. The response exhibits the classic non-minimum phase behavior. The step response has a rise time of around 2 seconds and is well damped. This can be categorized as a good response for the system.

Now, the objective is to look at this controller when the payload is dynamic (pendulum free to oscillate). Figure 4.8 shows what happens when the pendulum is given an initial dispacement while the controller is trying to regulate. The first plot in Figure 4.8 shows the time history of the pendulum. The response is stable. However, the pendulum damps only at its natural damping. The second plot in Figure 4.8 shows that the vibration of the payload is causing the endpoint to wander a few centimeters off the desired position.

This controller is unable to take energy out of the payload which leaves residual uncontrolled vibrations. The effect of the dynamics of the payload will certainly be even more dramatic if the endbody position is included as a feedback signal. Recall, Figure 4.3 showed a that the payload has a large effect on the endbody position. This kind of response would certainly be unsatisfactory for most space applications.

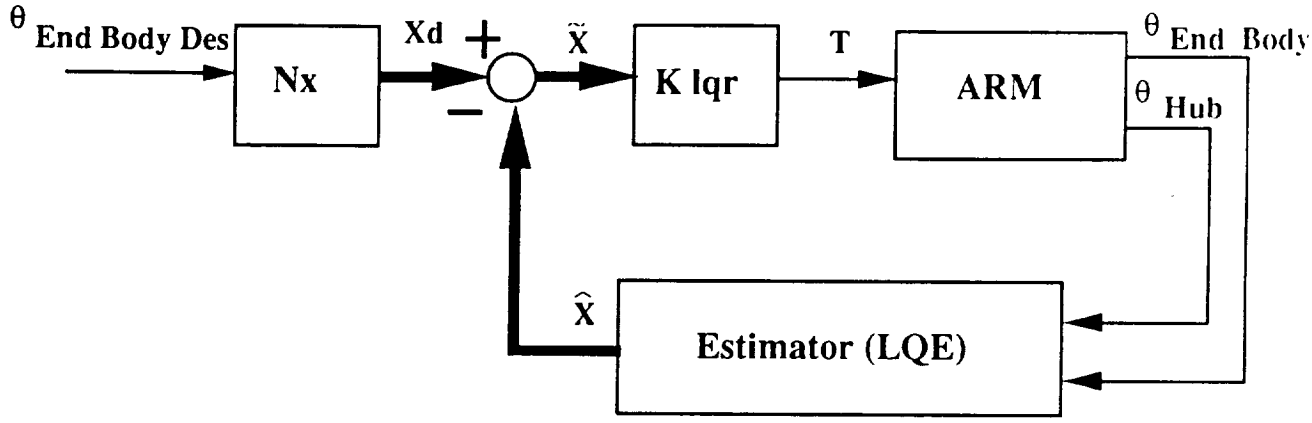

Figure 4.6: **Block Diagram**

4.0.7 Conclusions - Future Work

This paper has outlined the design and construction of a experimental test bed for examining the effects of payload dynamics on control of a flexible arm. It has been experimentally verified that a non-collocated controller has poor performance when the dynamics of the payload are ignored. The controller is shown to be unable to take energy out of the payload. The undamped vibrations of the payload cause poor regulation of tip position. This is all motivation for future study of the problem.

This paper has also shown that some signals are more sensitive than others to the dynamics of the payload. Future work includes, looking into the various sensor sets for control and identification. Some sensors such as the endbody angle, and hub angle will be good to use for robust control since they are insensitive to payload dynamics. Other sensors such as endbody position will be useful for adaptive schemes since they are very sensitive to the payload dynamics.

The far reaching goals are to position the payload in space and at the same time damp the internal vibrations of the payload without directly sensing or modelling the internal dynamics of the payload. This will certainly lead to studies of robust versus adaptive control, and sensor set trade-offs.

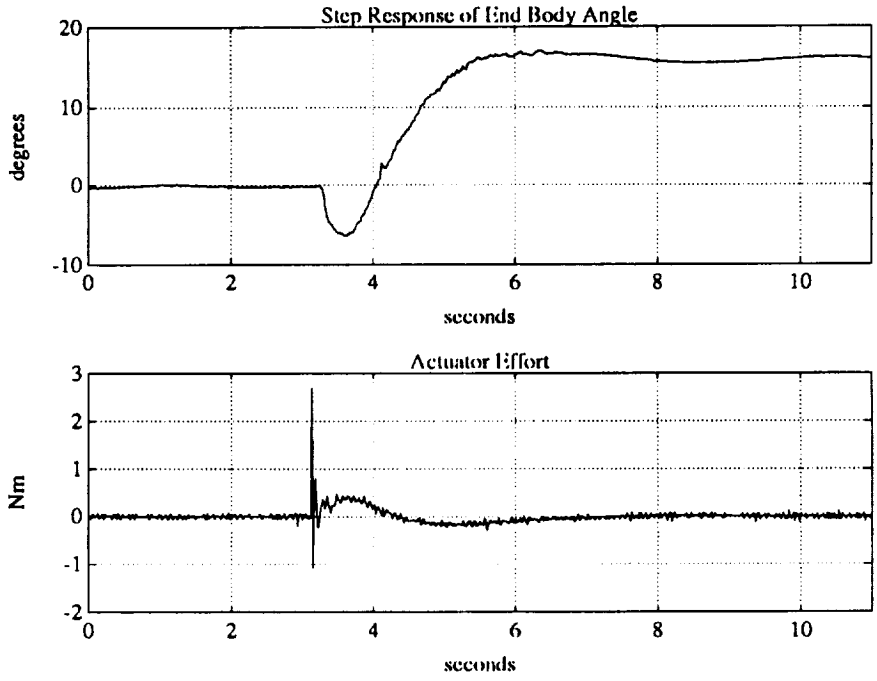

Figure 4.7: **Step Response in $\theta_{endbody}$** *non-collocated control: Static Payload*

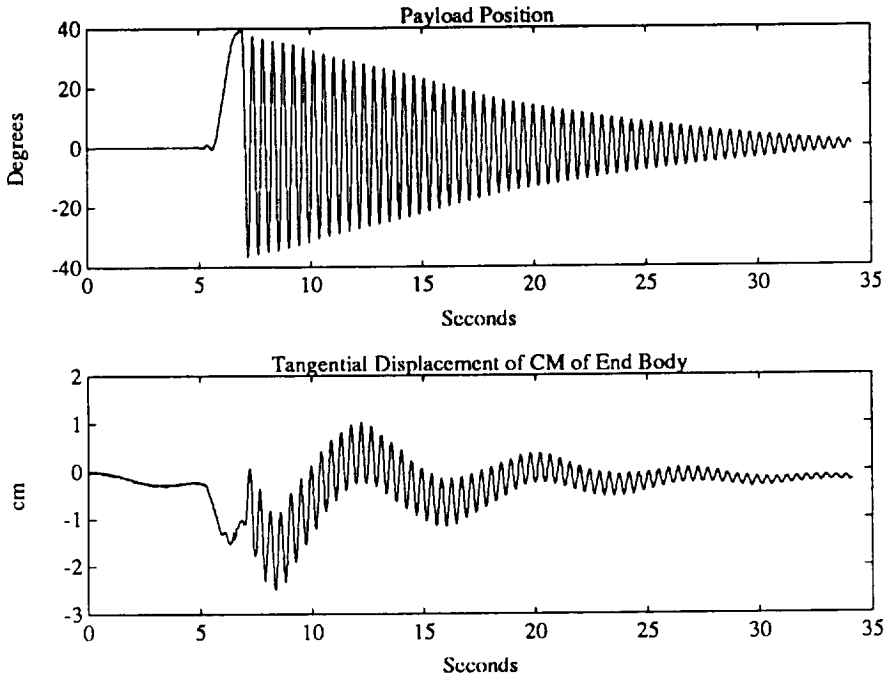

Figure 4.8: **Response to Dynamic Payload**

non-collocated control: Dynamic Payload

Chapter 5

Adaptive Neural Networks for Control of Space Robots

Edward Wilson

5.1 Introduction

Because they are capable of complex learned behaviors, adaptive neural networks have the potential to make a significant impact on the field of robotics in the near future. To investigate this potential, the ARL is launching a new program of experimental research to examine the applicability of this exciting technology to the control of space robot manipulator systems.

Neural networks are loosely modeled after the human brain. Instead of performing calculations sequentially on a single processor, calculations are performed simultaneously (even asynchronously) by a network of relatively simple processors. These processors act only locally, producing a single output based on a limited number of inputs (often the outputs of neighboring processors), just as the neurons in a human brain do.

Networks of these simple processors have emergent properties that allow very complex behavior–such as learning and pattern recognition–that are presently very limited in current computers. For example, neural networks may be used to implement arbitrary mappings of inputs to outputs, such as sensor signals to actuator commands. Since the mapping can be taught indirectly, neural networks are especially attractive for poorly-understood systems; they can generalize from training inputs and then respond in untaught situations. Due to the distributed nature of the processing, networks are often robust to internal component failures; the remaining processors can adapt to account for the failure. Similarly, the network can be made to adapt to changes in the environment, plant, performance criteria, etc.

One significant advantage of neural networks is that they may ultimately be implemented on parallel processing hardware for greatly enhanced throughput capabilities; however, they are often implemented on traditional sequential processing computers during

development, and when processing speed is not a limiting factor.

5.2 Possible Experimental Investigations

Our initial experiments will employ this strategy of using existing "sequential" computers. In particular, the processor on huey, the robot constructed by Warren Jasper for the thrusterless locomotion experiment, will be used.

We are investigating several possible applications. For example, a neural network could be used to calculate bias torques (due to friction, motor bias, wires, hoses, etc.), thus augmenting an existing controller. A neural network is well-suited to perform this bias torque mapping because:

- The problem requires some form of learning.
- Sources of the bias are not fully understood.
- The bias mapping is sure to be non-linear.
- The mapping could be time-varying (especially if the hoses and wires shift around) which would require some sort of on-line adaptation.

The mapping could be either valid over the entire workspace or simply used to "tweak" a single repeated maneuver (perhaps a jump from one end of the table to the other, or crawling along a railing) to remove trajectory errors.

In "supervisory learning", the parameters in a network are chosen by training it to emulate another controller. In a series of applications we could employ supervisory learning in training a network to emulate proportional-integral-derivative, bang-bang, computed torque, or even human controllers on an existing space robot manipulator system. This relatively simple training technique will yield important information about learning capabilities and the computational requirements for more sophisticated neural controllers.

Chapter 6

Multi-Sensor Fusion in a Space Robot

Kurt R. Zimmerman

6.1 Introduction

Multi-sensor fusion is redundancy of similar and/or dissimilar sensors to create a more robust control system. Expansion of this theory is critical to the success of robotic systems working in unconstrained environments. Multi-sensor fusion techniques may provide space robots with the increased reliability required to meet stringent space qualification demands, especially for robotic systems working in close proximity with humans in space. Graceful degredation is an important feature of systems employing multiple sensor redundancy since the loss of a single sensor will not result in failure of the entire system.

6.2 Multi-Sensor Fusion Techniques

A summary of common sensor fusion techniques can be found in [6]. The basic concepts are outlined here. Sensor data may come from similar sources (such as two ccd cameras viewing the same scene from two different vantage points) or disimilar sources (such as a ccd camera and a range-finder viewing the same scene from the same vantage point). In the case of dissimilar sources, the data sets must be transformed in a preprocessing step before fusing. The prominent techniques for combining the sensor data are averaging, guiding, and Bayesian statistics. Averaging is the simplest; the data is merely combined in a weighted manner where the weights are sensor confidence values. Guiding is the use of a simple, fast sensor to focus the attention of a slower, high resolution sensor. Bayesian statistical methods incorporate sensor uncertainty to determine the expected state of the environment. Kalman filtering of sensor data is an example of a Bayesian statistical approach.

6.3 Research Goals and Future Work

The primary goal of this research is to establish methods to improve the robustness of space robots through the use of multi-sensor fusion techniques. Two projects under consideration are:

- The use of multiple sensors to capture an unmodelled floating object: This project would be implemented on one of our two-arm model satellite robots. It would involve the use of multiple visual sensors guided by proximity sensors to obtain a nominal approximation of the shape of the object and tactile sensors guided by proximity sensors to capture the object. The resulting technology would prove beneficial in retrieving stray objects during construction of space structures and capturing space debris.

- Multi-robot assembly task as a multi-sensor fusion problem: A decentralized sensor fusion architecture as proposed in [5] would be implemented to coordinate several robots in a space assembly task. The architecture would be decentralized in that each sensor would be equipped with a processor to make its own estimation of state with a Kalman filter. The reason for using this approach is that when two or more robots are grasping the same object, the object and the robots can be considered one entity with multiple, redundant sensors. Since all sensors produce their own estimates, we can select the subset of sensors pertaining to the two robots grasping the object and fuse that data to establish the best estimate for the combined system. As configurations of robots and objects change, we can merely change the subsets of sensor data that are being fused in each situation.

- Also, projects involving sensor fusion through the use of neural networks are being considered. The parallel processing and interpolative nature of a neural network makes it an ideal candidate for processing multiple sensor information.

6.4 Conclusions

The usefulness of space robots can greatly be increased by making them more autonomous. However, higher levels of autonomy will require increased perception of the environment, which can be achieved by appropriately fusing data from multiple sensors. Experiments in sensor fusion will lead to better reliability of space robotic systems.

Bibliography

[1] Arthur E. Bryson, Jr. *Control of Spacecraft and Aircraft.* To be published.

[2] Robert H. Cannon, Jr., Marc Ullman, Ross Koningstein, William Dickson, Warren Jasper, and Lawrence Alder. NASA Semi-Annual Report on Control of Free-Flying Space Robot Manipulator Systems. Semi-Annual Report 9, Stanford University Aerospace Robotics Laboratory, Stanford, CA 94305, August 1990.

[3] Robert H. Cannon, Jr., Marc Ullman, Ross Koningstein, Stan Schneider, Warren Jasper, Roberto Zanutta, and William Dickson. NASA Semi-Annual Report on Control of Free-Flying Space Robot Manipulator Systems. Semi-Annual Report 7, Stanford University Aerospace Robotics Laboratory, Stanford, CA 94305, August 1988.

[4] P. Carton, J.P. Cretien, and M. Maurette. Simulatoin and control of space manipulators bearing complex payloads. In *Proceedings of XIth IFACS Symposium on Automatic Control in Aerospace*, 1989.

[5] B.Y.S. Rao H. F. Durrant-Whyte and H. Hu. Toward a fully decentralized architecture for multi-sensor data fusion. In *Proceedings of the International Conference on Robotics and Automation*, pages 1331 – 1336, April 1990.

[6] Jay K. Hackett and Mubarak Shah. Multi-sensor fusion: A perspective. In *Proceedings of the International Conference on Robotics and Automation*, pages 1324 – 1330, San Francisco, CA, April 1990. IEEE.

[7] Ross Koningstein. *Experiments in Cooperative-Arm Object Manipulation with a Two-Armed Free-Flying Robot.* PhD thesis, Stanford University, Department of Aeronautics and Astronautics, Stanford, CA 94305, October 1990. Also published as SUDAAR 597.

[8] Leonard Meirovitch. *Elements of Vibration Analysis.* Macmillan Series in Applied Mechanics. MacGraw-Hill Book Company, New York, NY, second edition, 1986.

[9] Daniel Mark Rovner. *Experiments in Adaptive Control of a Very Flexible One-Link Manipulator.* PhD thesis, Stanford University, Department of Aeronautics and Astronautics, Stanford, CA 94305, August 1987.

[10] Eric Schmitz. *Experiments on the End-Point Position Control of a Very Flexible One-Link Manipulator.* PhD thesis, Stanford University, Department of Aeronautics and Astronautics, Stanford, CA 94305, June 1985. Also published as SUDAAR 548.

~